# COMPUTER ARCHITECTURE

## A

## QUANTITATIVE

## APPROACH
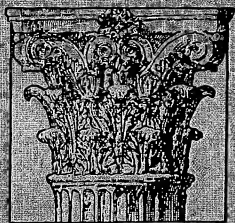
JOHN L. HENNESSY

&

DAVID A. PATTERSON

ADVICE, PRAISE, & ERRORS: Any correspondence related to this publication or
intended for the authors should be addressed to the editorial offices of Morgan Kaufmann
Publishers, Inc., Dept. P&H APE. Information regarding error sightings is encouraged.
Any error sightings that are accepted for correction in subsequent printings will be
rewarded by the authors with a payment of $1.00 (U.S.) per correction upon availability
of the new printing. Electronic mail can be sent to bugs2@vsop.stanford.edu. (Please
include your full name and permanent mailing address.)

INSTRUCTOR SUPPORT: For information on classroom software and other instructor
materials available to adopters, please contact the editorial offices of Morgan Kaufmann
Publishers, Inc. (415) 578-9911.

from the CPU. This type of organization is called a *memory hierarchy*. In Figure 1.6, a typical multilevel memory hierarchy is shown. Two important levels of the memory hierarchy are the cache and virtual memory.

A *cache* is a small, fast memory located close to the CPU that holds the most recently accessed code or data. When the CPU does not find a data item it needs in the cache, a *cache miss* occurs, and the data is retrieved from main memory and put into the cache. This usually causes the CPU to pause until the data is available.

Likewise, not all objects referenced by a program need to reside in main memory. If the computer has *virtual memory*, then some objects may reside on disk. The address space is usually broken into fixed-size blocks, called *pages*. At any time, each page resides either in main memory or on disk. When the CPU references an item within a page that is not present in the cache or main memory, a *page fault* occurs, and the entire page is moved from the disk to main memory. The cache and main memory have the same relationship as the main memory and disk.

| Level | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Called | Registers | Cache | Main memory | Disk storage |
| Typical size | < 1 KB | < 512 KB | < 512 MB | > 1 GB |
| Access time (in ns) | 10 | 20 | 100 | 20,000,000 |
| Bandwidth (in MB/sec.) | 800 | 200 | 133 | 4 |
| Managed by | Compiler | Hardware | Operating system | Operating system/user |
| Backed by | Cache | Main memory | Disk | Tape |

**FIGURE 1.7  The typical levels in the hierarchy slow down and get larger as we move away from the CPU.** Sizes are typical for a large workstation or minicomputer. The access time is given in nanoseconds. Bandwidth is given in MB per second, assuming 32-bit paths between levels in the memory hierarchy. As we move to lower levels of the hierarchy, the access times increase, making it feasible to manage the transfer less responsively. The values shown are typical in 1990 and will no doubt change over time.

| Machine | Register size | Register access time | Cache size | Cache access time |
|---|---|---|---|---|
| VAX-11/780 | 16 32-bit | 100 ns | 8 KB | 200 ns |
| VAXstation 3100 | 16 32-bit | 40 ns | 1 KB on chip, 64 KB off chip | 125 ns |
| DECstation 3100 | 32 32-bit integer; 16 64-bit floating point | 30 ns | 64 KB instruction; 64 KB data | 60 ns |

**FIGURE 1.8  Sizes and access times for the register and cache levels of the hierarchy vary dramatically among three different machines.**